y dynamics that combine order and disorder, ...
s, are capable of evolving while those that are purely
well away from the edge of chaos, cannot evolve. At the
capable of endless variety, novelty, surprise – in short,
trapped on local fitness peaks look stable and comfort-
aiting for destruction by other species following messier
ise conditions which generate the dynamics of the edge
rs only when the agents are numerous enough and richly
gents impose *conflicting constraints* on each other and it
l to the movement of the system.

hen, that the manner in which competitive selection
ons depends upon the internal dynamic of the evolving
e pattern of connections, the self-organising interaction,
ich it is composed. The fitness landscape is not a given
le evolutionary strategies for a system, which it searches
er driven by chance. Rather, the fitness landscape itself is
interaction between agents. The notion of fitness land-
mes a metaphor for the internal dynamic of a system, not
over which it travels in search of a fit position. These
etwork are the connections between its entities and these
ting constraints. The internal dynamic is thus one of
of conflicting constraints at the same time, a paradoxical
d competition at the same time. Notice how connection,
all essential requirements for the evolution of a system.
ontrol' of the evolution of the system, it is nevertheless
anner and the source of this control lies in the pattern of
is is a very important point, because it is the conflicting
fficient stability in a network at the edge of chaos.
of complex adaptive systems modellers are not confined to
he evolution of life. The complex adaptive system model
other phenomena too.

ions of homogeneous agents

of a complex adaptive system: namely a flock of birds.
d the flocking behaviour of birds with a computer pro-
ork of moving agents called Boids. Each Boid follows the

distance from other objects in the environment including

other Boids in the neighbourhood.
ceived centre of mass of the Boids in the neighbourhood.

cient to produce flocking behaviour. So, Boids, each inter-
mall number of others according to its own local rules of
mergent, coherent pattern for the whole system of Boids.

Pg 249

behaviour emerges for the whole population. Flocking is an attractor for a system in which entities follow the three rules given above.

Note how all agents follow the *same rules*. Each agent is the same as every other agent and there is no variation in the way they interact with each other. Emergence here is, therefore, not the consequence of non-average behaviour, as was the case with dissipative structures in the last section. Instead, emergence is the consequence of local interaction between agents. Unlike dissipative structures, and because of the postulated uniformity of behaviour, these simulations cannot spontaneously move, of their own accord, from one attractor to another. Instead, they stay always with one attractor and show no evolution.

However, more complicated simulations of complex adaptive systems do take account of differences in agents or classes of agents and different ways of interacting. These simulations do then show the capacity to move spontaneously from one attractor to another and to evolve new ones. This is demonstrated by the simulation called Tierra (Ray, 1992).

Simulating populations of interacting heterogeneous agents

Organic life utilises energy to organise matter and it evolves, developing more and more diverse forms, as organisms compete and co-operate with each other for light and food in geographic space. An analogy to this would be digital life in which central processing unit (CPU) time organises strings of digits (programs) in the space of computer memory. Computer programs are then used as the analogue of living organisms. Would digital life evolve as bit strings and interact and compete for CPU time?

This is the question explored by Ray (1992) in his simulation. In this simulation, Ray, the programmer, designs the first digital organism, which he calls a *creature*, consisting of 80 instructions on how to copy itself. The first creature is thus a string of digits of a particular length. The programmer also introduces a mechanism to generate variety into the replicating process, taking the form of random bit flipping to simulate random mutations in evolution. It follows that, as the creature copies itself, the new copies will differ from the original one and, as they copy themselves, each subsequent copy will differ from them. The programmer also introduces a constraint in the form of scarce computer time, which works as follows. Agents are required to post their locations in the computer memory on a public notice board. Each agent is then called upon in turn, according to a circular queue, to receive a slice of computer time for carrying out its replication tasks. The programmer introduces a further constraint on agent life span. Agents are lined up in a linear queue according to their age and a 'reaper' lops off some of these, generally the oldest. However, by successfully executing their programs, agents can slow down their move up the linear queue, whereas flawed agents rise quickly to the top.

The only task agents have is that of replicating in a regime of scarce CPU time and what happens is that new modes of doing this evolve. In other words, different categories of replication method appear. These changes can be observed in numerical terms by watching changing patterns of dots on a computer screen. An analogy is then drawn between this digital interaction and the biological evolution of species and the simulation is described in these biological terms. For example, categories of agents are said to develop their own survival strategies. It is important to remember that this is an analogy drawing attention to changes in categories of agent in the digital medium and changes in categories of species in the biological medium.

What happens in the simulation?

The simulation was set off by introducing a single agent consisting of 80 instructions. Within a short time, the computer memory space was 80 per cent occupied by these agents but then the reaper took over and prevented further population growth. After a while, agents consisting of 45 instructions appeared, but they were too short to replicate. They overcame this problem by borrowing some of the code of longer agents in order to replicate. This strategy enabled them to replicate faster within their allocated computer time. In other words, a kind of parasite emerged. The use of the term 'parasite' is obviously an analogy.

Although the parasites did not destroy their hosts, they were dependent on them for replication. If the parasites became too numerous in relation to hosts, they destroyed their own ability to replicate and so declined. In the simulation, the parasites suffered periodic catastrophes. One of these catastrophes occurred because the hosts stopped posting their positions on the public notice board and in effect hid so that the parasites could no longer find them. Some hosts had, thus, developed an immunity to parasites by using camouflage as a survival strategy. On the other hand, in hiding, the hosts had not retained any note of their position in the computer memory. So, they had to examine themselves to see if their position corresponded to the position being offered computer time, before they could respond to that offer. This increased the time they needed for replication. However, although not perfect, the strategy worked well enough that the parasites were nearly wiped out.

Then, however, the parasites developed their own memories and did not need to consult the public posting board. Once again, it was the parasites' turn to succeed. Later, hyperparasites appeared to feed off the parasites. These were 80 instructions long, just like the hosts, but they had developed instructions to examine themselves for parasites and feed off the parasites by diverting computer time from them. These hyperparasites worked symbiotically by sharing reproduction code: they could no longer reproduce on their own but required cooperation. This cooperation was then exploited by opportunistic mutants in the form of tiny intruders who placed themselves between replicating hyperparasites and intercepted and used hyperparasite code for their own replication. These cheaters could then thrive and replicate although they were only 27 instructions long. Later, the hyperparasites found a way to defeat the cheaters, but not for long.

How the simulation is interpreted

I would like to emphasise, once more, what is happening in this simulation. After

replicate in a different way. In comple
attractor and there are a number of d
another way, there is micro diversity i
one round of replication – that is, duri
carry out their instructions, one after t
the strings are randomly flipped. Over
action between the bit strings result in
In other words, new arrangements of
replicating instructions. At the same ti
procedure of competitive removal of
continues even when the random bit
organisation is then the driving force

In summary, the population of bit st
procedures. What running the simul
of iteration (replication) and local in
the absence of a blueprint for the wh
competitive selection. The simulation
organisation, mutation and selection o
is, emergent novelty that is radically
by both destruction of some categorie

Anything more that is said about
analogy. So, Ray uses the simulation
strings *creatures*. One category of bi
parasites. If the interpretation is done
it may indicate that new biological
organisation, not just by chance. If d
claims. It is, therefore, important to
organisation and emergence in relat
one of how to interpret, in organisa
interaction between replicating algor
properties. Even more fundamental i
to try to do this.

Some major insights

It seems to me that this simulation p
complex adaptive systems.

First, this system produces evolvin
spontaneous, emergent way throug
evolving population-wide order has
grand design or plan for it. Further
with its emergent order, is vital fo
its ability to produce novelty. How
population-wide pattern of behavio

nalogy drawing attention to changes in categories of
ınd changes in categories of species in the biological

lation?

' introducing a single agent consisting of 80 instruc-
e computer memory space was 80 per cent occupied
reaper took over and prevented further population
consisting of 45 instructions appeared, but they were
'ercame this problem by borrowing some of the code
plicate. This strategy enabled them to replicate faster
er time. In other words, a kind of parasite emerged.
is obviously an analogy.
ıot destroy their hosts, they were dependent on them
es became too numerous in relation to hosts, they
o replicate and so declined. In the simulation, the
strophes. One of these catastrophes occurred because
positions on the public notice board and in effect hid
onger find them. Some hosts had, thus, developed an
camouflage as a survival strategy. On the other hand,
etained any note of their position in the computer
ıine themselves to see if their position corresponded
mputer time, before they could respond to that offer.
eded for replication. However, although not perfect,
gh that the parasites were nearly wiped out.
; developed their own memories and did not need to
'd. Once again, it was the parasites' turn to succeed.
to feed off the parasites. These were 80 instructions
ey had developed instructions to examine themselves
rasites by diverting computer time from them. These
ically by sharing reproduction code: they could no
out required cooperation. This cooperation was then
ants in the form of tiny intruders who placed them-
erparasites and intercepted and used hyperparasite
ı. These cheaters could then thrive and replicate
ructions long. Later, the hyperparasites found a way
'or long.

reted

: more, what is happening in this simulation. After
me time there are a number of bit strings, each
ions requiring them to replicate in a particular way,
bit strings. These bit strings fall into categories and
the same way, while bit strings in another category

Pg 251

another way, there is micro diversity in the total population of bit strings. During one round of replication – that is, during a given short time period – the bit strings carry out their instructions, one after the other, and as they do so bits in some of the strings are randomly flipped. Over a series of runs the bit flipping and the interaction between the bit strings result in rearrangements in the bit strings themselves. In other words, new arrangements of bit strings appear: that is, new categories of replicating instructions. At the same time older categories disappear because of the procedure of competitive removal of some of them. Once begun, this evolution continues even when the random bit flipping, that is, chance, is turned off. Self-organisation is then the driving force of evolution.

In summary, the population of bit strings is a population of algorithms, or logical procedures. What running the simulation demonstrates is the logical properties of iteration (replication) and local interaction of algorithms (self-organisation in the absence of a blueprint for the whole) in the presence of random mutation and competitive selection. The simulation shows that it is logically possible for self-organisation, mutation and selection operating iteratively to display evolution – that is, emergent novelty that is radically unpredictable. This evolution is characterised by both destruction of some categories and emergence of new ones.

Anything more that is said about the simulation is an interpretation by way of analogy. So, Ray uses the simulation as an analogy for biology and calls the bit strings *creatures*. One category of bit strings is called *hosts* and another is called *parasites*. If the interpretation is done carefully, it may provide insight. For example, it may indicate that new biological forms can emerge from a process of self-organisation, not just by chance. If done carelessly, it could produce unwarranted claims. It is, therefore, important to take great care in using insights about self-organisation and emergence in relation to organisations. The question becomes one of how to interpret, in organisational terms, the logic of iterative, nonlinear interaction between replicating algorithms and their self-organising and emergent properties. Even more fundamental is the question of whether it even makes sense to try to do this.

Some major insights

It seems to me that this simulation provides some major insights into the nature of complex adaptive systems.

First, this system produces evolving population-wide order that comes about in a spontaneous, emergent way through the local interaction of diverse agents. The evolving population-wide order has not been programmed and there is no blueprint, grand design or plan for it. Furthermore, this spontaneous self-organising activity, with its emergent order, is vital for the continuing evolution of the system and its ability to produce novelty. However, what form that order takes – that is, the population-wide pattern of behaviour, the system-wide strategies – cannot be predicted from the rules driving individual agent behaviour. The strategies are emerging unpredictably in co-evolutionary processes. First the strategy is small size, but then parasites change the rules and the most successful strategy becomes feeding off others. Then, the hosts change the rules and the better strategy is camouflage. But